**44-542 Object Oriented Programming**

Maps and Sets

**Overview**: For this exercise, we are going to work with an input file that comes in two parts that are handled differently. The first line in the file contains a list of words. The rest of the file contains text that you will analyze. Your task is to count the number of times each of the given words occurs in the text. You will use Sets and Maps to get the desired output.

For example, if the input file contains the following text:

**the sensor stomach pear**

**Once the ingestible sensor reaches the stomach, it is powered by contact with stomach fluid**

The output must be (order may vary):

**{the=2,sensor=1,stomach=1, pear=0}**

Notice that case and punctuation will matter. We only count one occurrence of "**stomach**" because "**stomach,**" is not the same. Handling punctuation correctly is fairly tricky, so we will just use the standard Scanner and break on white space. An expanded example is given below:

Sample Input:

|  |
| --- |
| The sensor or products identity on metrics has the sensor product as unidentified on several metrics  The Proteus ingestible sensor can be integrated into an inert pill or other ingested products, such as pharmaceuticals. Once the ingestible sensor reaches the stomach, it is powered by contact with stomach fluid and communicates a unique signal that determines identity and timing of ingestion. This information is transferred through the user’s body tissue to a patch worn on the skin that detects the signal and marks the precise time an ingestible sensor has been taken. Additional physiologic and behavioral metrics collected by the patch include heart rate, body position and activity. The patch relays information to a mobile phone application. With the patient’s consent, the information is accessible by caregivers and clinicians, helping individuals to develop and sustain healthy habits, families to make better health choices, and clinicians to provide more effective, data-driven care. |

Sample Output:

|  |
| --- |
| {The=2, the=9, as=1, or=1, identity=1, sensor=3, has=1, metrics=1, on=1} |

Note: One useful trick is that we can create a Scanner and pass in a string to be parsed. Example:

**Scanner lineScanner = new Scanner("Some string of text");**